1.

2.

$$err(\mathbf{w}) = \begin{cases} 0, & -y\,\mathbf{w}^T\mathbf{x} \le 0 \,(correct\ classified) \\ -y\mathbf{w}^T\mathbf{x}, & -y\,\mathbf{w}^T\mathbf{x} > 0\,(incorrect\ classified) \end{cases}$$

Here we only consider the case that some points are classified wrong, meaning that we only discuss cases that make $err(\mathbf{w}) = -y\mathbf{w}^T\mathbf{x}$.

$$\frac{\partial(-y\mathbf{w}^T\mathbf{x})}{\partial w_i} = -y\,x_i$$

While w_i stands for the i^{th} element in vector $\mathbf{w}$, x_i stands for the i^{th} element in vector $\mathbf{x}$.

So, the gradient of $-y\mathbf{w}^T\mathbf{x}$ is:

$$\nabla(-y\mathbf{w}^T\mathbf{x}) = -y\mathbf{x}$$

Therefore, for a point that is classified wrong, according to the update rules of SGD,

$$\mathbf{w}(\boldsymbol{t}+\mathbf{1}) = \mathbf{w}(\boldsymbol{t}) - \eta\nabla(-y\mathbf{w}^T\mathbf{x}) = \mathbf{w}(\boldsymbol{t}) + \eta y\mathbf{x}, \eta\ is\ the\ step\ size$$

While the update rule of PLA looks like this when a point is classified wrong:

$$\mathbf{w}(\boldsymbol{t}+\mathbf{1}) = \mathbf{w}(\boldsymbol{t}) + y\mathbf{x}$$

So when $\eta = 1$, $err(\mathbf{w}) = \max(0, -y\,\mathbf{w}^T\mathbf{x})$, using SGD will result in PLA.

3.

$$E_{in} = \frac{1}{N}\sum_{n=1}^{N}\left(\ln\left(\sum_{i=1}^{K}\exp\left(\mathbf{w}_i^T\mathbf{x}_n\right)\right) - \mathbf{w}_{y_n}^T\mathbf{x}_n\right)$$

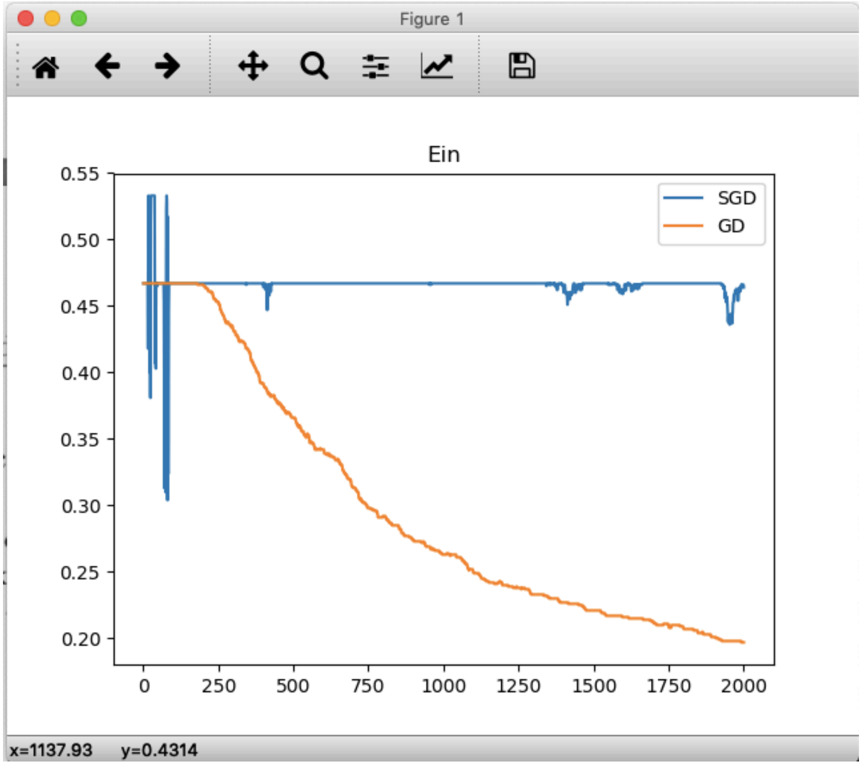
$$\frac{\partial E_{in}}{\partial \mathbf{w}_i} = \frac{1}{N}\sum_{n=1}^{N}\left(\frac{\partial\left(\ln\left(\sum_{i=1}^{K}\exp\left(\mathbf{w}_i^T\mathbf{x}_n\right)\right)\right)}{\partial \mathbf{w}_i} - \frac{\partial\left(\mathbf{w}_{y_n}^T\mathbf{x}_n\right)}{\partial \mathbf{w}_i}\right)$$

= 1□N□ n = 1□N□ 1□ i =

1□K□ exp□ 𝒘□𝒊□𝑻□ 𝒙□𝒏□□□□□ · exp□ 𝒘□𝒊□𝑻□ 𝒙□𝒏□□ · 𝒙□𝒏□□ − y□n□ = i□ · 𝒙□𝒏□ □□

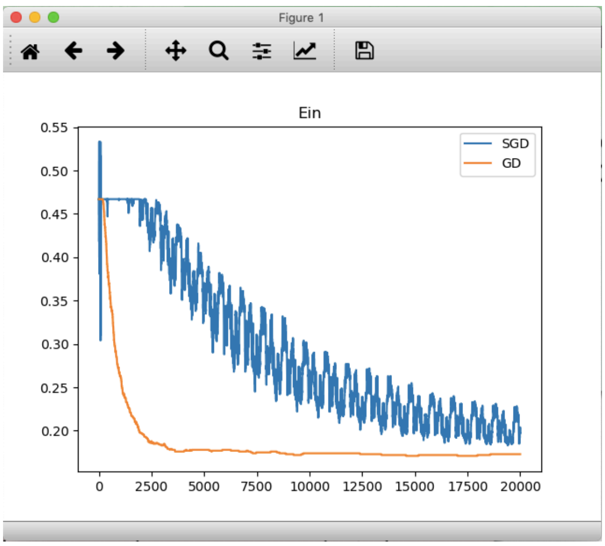
= 1□N□ n = 1□N□ (h□i□ 𝒙□𝒏□□ − y□n□ = i□) · 𝒙□𝒏□ □□

4.



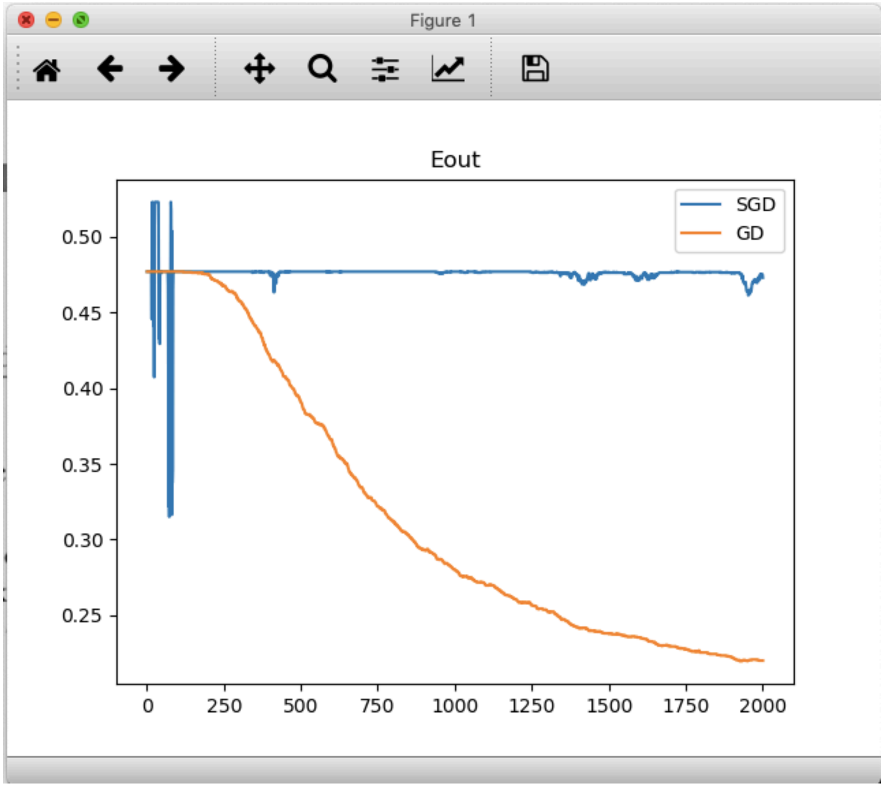
SGD stands for Stochastic Gradient Descent, GD stands for Gradient Descent

We can see that the Ein of GD decrease steadily, which is reasonable because GD calculates the 'true' gradient so the Ein of GD should go down steadily. SGD, on the other hand, calculates the gradient on one point to approximate the 'true' gradient. This causes the Ein of SGD to not decrease steadily, because sometimes the 'approximate' gradient is far away from the true gradient. However, after enough iterations, SGD still can reach the result of GD. In fact, I changed the iterations to 20000 to test my hypothesis. Here's the result.



For 2500~20000 iterations, the Ein of GD almost doesn't change, so I think there's some noise in the data.

5.



SGD stands for Stochastic Gradient Descent, GD stands for Gradient Descent

The Eout graph is similar to the Ein graph, but there are still some differences. After 2000 iterations, Eout of GD is close but slightly greater than Ein of GD. This is reasonable because of VC Bound which tells us that the distance between Ein and Eout will not be too far.

6.

Let $\boldsymbol{h} = \begin{bmatrix} h_1(\boldsymbol{x_1}) & h_2(\boldsymbol{x_1}) & \cdots & h_K(\boldsymbol{x_1}) \\ h_1(\boldsymbol{x_2}) & \ddots & & \vdots \\ \vdots & & \ddots & h_K(\boldsymbol{x_{N-1}}) \\ h_1(\boldsymbol{x_N}) & \cdots & h_{K-1}(\boldsymbol{x_N}) & h_K(\boldsymbol{x_N}) \end{bmatrix}, \boldsymbol{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_K \end{bmatrix}, \boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}, \boldsymbol{x} = \begin{bmatrix} \boldsymbol{x_1} \\ \boldsymbol{x_2} \\ \vdots \\ \boldsymbol{x_N} \end{bmatrix}$

$$RMSE(H) = \sqrt{\frac{1}{N}\sum_{n=1}^{N}\left(y_n - \sum_{k=1}^{K} w_k h_k(\boldsymbol{x_n})\right)^2}$$

$$= \sqrt{\frac{1}{N}\left\|\boldsymbol{y} - \boldsymbol{hw}\right\|^2}$$

$$\operatorname*{argmin}_{\boldsymbol{w}} RMSE(H) = \operatorname*{argmin}_{\boldsymbol{w}} \left\|\boldsymbol{y} - \boldsymbol{hw}\right\|^2 = \operatorname*{argmin}_{\boldsymbol{w}} (\boldsymbol{y}^T\boldsymbol{y} - 2\boldsymbol{w}^T\boldsymbol{h}^T\boldsymbol{y} + \boldsymbol{w}^T\boldsymbol{h}^T\boldsymbol{hw})$$

Let $err(\boldsymbol{w}) = (\boldsymbol{y}^T\boldsymbol{y} - 2\boldsymbol{w}^T\boldsymbol{h}^T\boldsymbol{y} + \boldsymbol{w}^T\boldsymbol{h}^T\boldsymbol{hw})$, the gradient of err is:

$$\nabla err(\boldsymbol{w}) = 2\boldsymbol{h}^T\boldsymbol{hw} - 2\boldsymbol{h}^T\boldsymbol{y}$$

To minimize $err(\boldsymbol{w})$, $\nabla err(\boldsymbol{w})$ should be zero, therefore:

$$\boldsymbol{w} = (\boldsymbol{h}^T\boldsymbol{h})^{-1}\boldsymbol{h}^T\boldsymbol{y}$$

Now, we need to get $\boldsymbol{h}^T\boldsymbol{y}$ to get $\boldsymbol{w}$.

How do we get $\boldsymbol{h}^T\boldsymbol{y}$?

$$\boldsymbol{h}^T\boldsymbol{y} = \begin{bmatrix} h_1(\boldsymbol{x}_1) & h_1(\boldsymbol{x}_2) & \cdots & h_1(\boldsymbol{x}_N) \\ h_2(\boldsymbol{x}_1) & \ddots & & \vdots \\ \vdots & & \ddots & h_{K-1}(\boldsymbol{x}_N) \\ h_K(\boldsymbol{x}_1) & \cdots & h_K(\boldsymbol{x}_{N-1}) & h_K(\boldsymbol{x}_N) \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$

Let $\boldsymbol{r}_{\boldsymbol{i}}$ be the i_{th} row of $\boldsymbol{h}^{\boldsymbol{T}}$, and

$$\boldsymbol{r}_{\boldsymbol{0}} = \begin{bmatrix} h_0(\boldsymbol{x}_{\boldsymbol{1}}) & h_0(\boldsymbol{x}_{\boldsymbol{2}}) & \cdots & h_0(\boldsymbol{x}_{\boldsymbol{N}}) \end{bmatrix} = \begin{bmatrix} 0 & 0 & \cdots & 0 \end{bmatrix}$$

From the problem statement,

$$e_i = \frac{1}{N}\sum_{n=1}^{N}\left(y_n - h_i(\boldsymbol{x}_{\boldsymbol{n}})\right)^2 = \frac{1}{N}\left\|\boldsymbol{y} - \boldsymbol{r}_{\boldsymbol{i}}^{\boldsymbol{T}}\right\|^2$$

$$N e_i^2 = \left\|\boldsymbol{y} - \boldsymbol{r}_{\boldsymbol{i}}^{\boldsymbol{T}}\right\|^2 = \boldsymbol{y}^T\boldsymbol{y} - 2\,\boldsymbol{r}_{\boldsymbol{i}}\boldsymbol{y} + \boldsymbol{r}_{\boldsymbol{i}}\boldsymbol{r}_{\boldsymbol{i}}^{\boldsymbol{T}}$$

For any $j \in [1, K]$,

$$N\left(e_0^2 - e_1^2\right) = 2\left(\boldsymbol{r}_{\boldsymbol{1}} - \boldsymbol{r}_{\boldsymbol{0}}\right)\boldsymbol{y} + \boldsymbol{r}_{\boldsymbol{0}}\boldsymbol{r}_{\boldsymbol{0}}^{\boldsymbol{T}} - \boldsymbol{r}_{\boldsymbol{1}}\boldsymbol{r}_{\boldsymbol{1}}^{\boldsymbol{T}}$$

$$N\left(e_1^2 - e_2^2\right) = 2\left(\boldsymbol{r}_{\boldsymbol{2}} - \boldsymbol{r}_{\boldsymbol{1}}\right)\boldsymbol{y} + \boldsymbol{r}_{\boldsymbol{1}}\boldsymbol{r}_{\boldsymbol{1}}^{\boldsymbol{T}} - \boldsymbol{r}_{\boldsymbol{2}}\boldsymbol{r}_{\boldsymbol{2}}^{\boldsymbol{T}}$$

$$\vdots$$

$$N\left(e_{j-1}^2 - e_j^2\right) = 2\left(\boldsymbol{r}_{\boldsymbol{j}} - \boldsymbol{r}_{\boldsymbol{j-1}}\right)\boldsymbol{y} + \boldsymbol{r}_{\boldsymbol{j-1}}\boldsymbol{r}_{\boldsymbol{j-1}}^{\boldsymbol{T}} - \boldsymbol{r}_{\boldsymbol{j}}\boldsymbol{r}_{\boldsymbol{j}}^{\boldsymbol{T}}$$

Add them all, we get:

$$N\left(e_0^2 - e_j^2\right) = 2\left(\boldsymbol{r}_{\boldsymbol{j}} - \boldsymbol{r}_{\boldsymbol{0}}\right)\boldsymbol{y} + \boldsymbol{r}_{\boldsymbol{0}}\boldsymbol{r}_{\boldsymbol{0}}^{\boldsymbol{T}} - \boldsymbol{r}_{\boldsymbol{j}}\boldsymbol{r}_{\boldsymbol{j}}^{\boldsymbol{T}} = 2\,\boldsymbol{r}_{\boldsymbol{j}}\boldsymbol{y} - \boldsymbol{r}_{\boldsymbol{j}}\boldsymbol{r}_{\boldsymbol{j}}^{\boldsymbol{T}} - 2\,\boldsymbol{r}_{\boldsymbol{0}}\boldsymbol{y} + \boldsymbol{r}_{\boldsymbol{0}}\boldsymbol{r}_{\boldsymbol{0}}^{\boldsymbol{T}}$$

Notice that $\boldsymbol{r}_{\boldsymbol{0}}\boldsymbol{y} = \begin{bmatrix} 0 & 0 & \cdots & 0 \end{bmatrix}$, $\boldsymbol{r}_{\boldsymbol{0}}\boldsymbol{r}_{\boldsymbol{0}}^{\boldsymbol{T}} = \begin{bmatrix} 0 & 0 & \cdots & 0 \end{bmatrix}$, So:

$$N e_0^2 - N e_j^2 = 2\,\boldsymbol{r}_{\boldsymbol{j}}\boldsymbol{y} - \boldsymbol{r}_{\boldsymbol{j}}\boldsymbol{r}_{\boldsymbol{j}}^{\boldsymbol{T}}$$

$$\boldsymbol{r}_{\boldsymbol{j}}\boldsymbol{y} = \frac{1}{2}\left(N e_0^2 - N e_j^2 + \boldsymbol{r}_{\boldsymbol{j}}\boldsymbol{r}_{\boldsymbol{j}}^{\boldsymbol{T}}\right)$$

Notice that $\boldsymbol{h}^T\boldsymbol{y} = \begin{bmatrix} h_1(\boldsymbol{x}_1) & h_1(\boldsymbol{x}_2) & \cdots & h_1(\boldsymbol{x}_N) \\ h_2(\boldsymbol{x}_1) & \ddots & & \vdots \\ \vdots & & \ddots & h_{K-1}(\boldsymbol{x}_N) \\ h_K(\boldsymbol{x}_1) & \cdots & h_K(\boldsymbol{x}_{N-1}) & h_K(\boldsymbol{x}_N) \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} \boldsymbol{r}_{\boldsymbol{1}} \\ \boldsymbol{r}_{\boldsymbol{2}} \\ \vdots \\ \boldsymbol{r}_{\boldsymbol{K}} \end{bmatrix}\boldsymbol{y} = \sum_{k=1}^{K}\boldsymbol{r}_{\boldsymbol{k}}\boldsymbol{y}$

So, we now know that $\boldsymbol{h}^T\boldsymbol{y} = \sum_{k=1}^{K}\boldsymbol{r}_{\boldsymbol{k}}\boldsymbol{y} = \sum_{k=1}^{K}\frac{1}{2}\left(N e_0^2 - N e_k^2 + \boldsymbol{r}_{\boldsymbol{k}}\boldsymbol{r}_{\boldsymbol{k}}^{\boldsymbol{T}}\right) = \sum_{k=1}^{K}\frac{1}{2}\left(N e_0^2 - N e_k^2 + \sum_{\boldsymbol{n}=\boldsymbol{1}}^{\boldsymbol{N}}\left(h_k(\boldsymbol{x}_{\boldsymbol{n}})\right)^{\boldsymbol{2}}\right)$

By and large, when $\boldsymbol{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_K \end{bmatrix} = \left(\boldsymbol{h}^T\boldsymbol{h}\right)^{-1}\boldsymbol{h}^T\boldsymbol{y} = \left(\begin{bmatrix} h_1(\boldsymbol{x}_1) & h_1(\boldsymbol{x}_2) & \cdots & h_1(\boldsymbol{x}_N) \\ h_2(\boldsymbol{x}_1) & \ddots & & \vdots \\ \vdots & & \ddots & h_{K-1}(\boldsymbol{x}_N) \\ h_K(\boldsymbol{x}_1) & \cdots & h_K(\boldsymbol{x}_{N-1}) & h_K(\boldsymbol{x}_N) \end{bmatrix} \begin{bmatrix} h_1(\boldsymbol{x}_1) & h_2(\boldsymbol{x}_1) & \cdots & h_K(\boldsymbol{x}_1) \\ h_1(\boldsymbol{x}_2) & \ddots & & \vdots \\ \vdots & & \ddots & h_K(\boldsymbol{x}_{N-1}) \\ h_1(\boldsymbol{x}_N) & \cdots & h_{K-} \end{bmatrix}\right.$

$$1□\ \boldsymbol{x}□\boldsymbol{N}□□□\ h□K□\ \ \boldsymbol{x}□\boldsymbol{N}□□□□□□ - \mathbf{1}□\ k = 1□K□\ \ 1□2□\ N\ e□0□2□ - N\ e□k□2□ + \ \boldsymbol{n} =$$

$$\mathbf{1}□\boldsymbol{N}□\ \ \ h□k□\ \ \boldsymbol{x}□\boldsymbol{n}□□□□□\mathbf{2}□□□□□,$$

$$RMSE\ H□\ will\ achieve\ its\ minimum\ value.$$